The Planetary Data System
Distributed Inventory System

J. Steven Hughes
Steve.Hughes@jpl.nasa.gov
and
Susan K. McMahon
Susan.McMahon@jpl.nasa.gov

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA

Abstract

The advent of the World Wide Web (Web) and the ability to easily put data repositories on-line has resulted in a proliferation of digital libraries. The heterogeneity of the underlying systems, the autonomy of the individual sites, and distributed nature of the technology has made both interoperability across the sites and the search for resources within a site major research topics. This article will describe a system that addresses both issues using standard Web protocols and meta-data labels to implement an inventory of on-line resources across a group of sites. The success of this system is strongly dependent on the existence of and adherence to a standards architecture that guides the management of meta-data within participating sites.

1.0 Introduction

The Planetary Data System (PDS) [1] is an active science data archive managed by scientists for NASA's planetary science community and has been in operation since 1990. Envisioned as a long-term archive, the PDS early on emphasized the development of a standards architecture that would include both the science data and the meta-data necessary for interpreting diverse data storage formats as well as understanding the context under which the data were captured. This standards architecture has been used to create a high quality, peer-reviewed, science data archive of about five terabytes that is stored on Compact Disk (CD) media. The meta-data in this archive, even though collected to ensure the usability of the science data for future scientists, has also allowed the majority of the archive to be made available on-line via the Web as a digital library. As a component of this digital library, the Distributed Inventory System (DIS) was developed to identify all resources across the heterogeneous, autonomous, and distributed nodes of the PDS. This inventory includes all archived data sets, pending data sets, and any resources that support the use of data sets. The DIS is a lightweight solution to managing access to PDS resources, taking advantage of the wealth of meta-data available in the archive.

In the following we will give a brief history of the PDS and the original approach for managing its resources. This will be followed by a detailed description of the Distributed Inventory System (DIS). Future directions for development, focusing on current digital library research and integration with other science data systems will conclude the paper.

2.0 Background

In 1983, the Committee on Data Management and Computing (CODMAC) issued a report that set the guidelines for the development of a science data archive. [2] This report resulted from the observation that a wealth of science data would ultimately cease to be useful and probably lost if a process was not developed to ensure that the science data were properly archive. In particular, the report proposed two major goals. First, the committee recommended that the data be transferred to more stable media. Second, the committee recommended that sufficient ancillary and meta-data be captured and archived with the data to ensure that future users of the data would be able to understand how to interpret the data formats as well as understand the context under which the data was collected and processed.

Additionally, the report suggested that sophisticated searches for data sets were important. In particular the scientists wanted the capability to find data sets through relationships with other entities. For example, using the relationships between data sets, spacecraft, and instruments, scientists wanted the ability to identify images that had been captured using specific filters on a specific camera type on any spacecraft.

The PDS was organized in 1985 and five years of design and development followed. It went on-line in 1990 with about 20 data sets in its archive. A major component of the system was a high-level data set catalog that allowed the searching and ordering of any data set in the archive through an on-line interface of about 88 user views. To ingest data into the archive, the PDS developed a data ingestion procedure that includes a formal peer review. This review involves a peer review committee comprised of peer scientists who review the science data for validity and usability and technical staff that review the collected science data and meta-data for adherence to the standards architecture. The PDS today has about 400 peer-reviewed data sets in its archive. Another 100 are pending peer review. The standards architecture and procedures, including the peer review, are documented in three volumes, the Data Preparation Workbook [3], the Standards Reference [4], and the Planetary Science Data Dictionary (PSDD) [5]. The PDS data model and the meta-data development and management process have been described in [6] and [7].

Within the planetary science community an example of a science data set is the collection of about 50,000 Mars images returned by the Viking Orbiter spacecraft in 1976. An individual image within this data is an example of a data set granule or product.

3.0 The PDS Approach to Managing Its Data

The original vision for managing PDS data focused on adhering to a standards architecture, distributing the data for management by discipline scientists, and allowing access to the data through an integrated set of catalogs.

The PDS is a distributed system, consisting of five science discipline nodes, two support nodes and a central node. Contracted out in most cases to universities, the discipline nodes manage the science data within their discipline while also providing science expertise. Originally, each node was solely responsible for distributing their data. The central node managed the data set catalog and forwarded orders to the discipline nodes. The data were distributed on CD or on tape.

The standards architecture is the cohesive element of the PDS, allowing the autonomous and distributed science discipline nodes to produce archive volumes that are very much alike in organization, quality, and usability.

With the advent of the Web and a growing volume of data to ingest into the archive, the PDS had to evolve. In particular, Web technology has allowed advanced on-line interfaces to be developed in a fraction of the time that was previously required. This allowed better access to the archive also but resulted in more users. Also, additional copies of the data could be put on-line at other sites with relative ease. This addressed bandwidth problems but required more management because of redundancy. Finally, users wanted access to data sets that were not fully archived. Since only archived data sets were ingested into the data set catalog, pending data sets were accessible only to a select few that were told where to find the data.

This combination of new technology and the need to manage multiple on-line copies of archived and pending data sets and non-data resources, led to the collection of requirements for an inventory system. Limited resources dictated a lightweight solution focusing on simple and automatic operations.

4.0 Purpose of the DIS

There were originally three purposes for the DIS. First, it was to improve user access to the PDS by identifying what data sets and resources were available either on or off-line at the time of the query. To accomplish this, an inventory was to be taken across all nodes on a regular basis to track the location of all copies of a data set, verify the on-line status on a regular basis, and check Web site status. This system was to build on the existing data set catalog of archived products.

The second purpose was to help the user by clarifying the quality of the available data sets. Since the ingestion process can take months and in some cases years to complete, it

was important to notify the user of the quality of the data. To accomplish this, the archive_status and archive_status_note attributes were created.

The third purpose was to provide disaster recovery. With a complete inventory of all data sets and resources resident in a single database, a backup copy of the database could be made and the inventory could be brought up at an alternate site.

5.0 The DIS Approach

The Distributed Inventory System (DIS) consists of two major components, the central DIS manager and individual DIS component at each of the nodes. All the components are conceptually identical. The DIS manager contains the inventory across the entire PDS while a node component contains the inventory local to that node. The inventory consists primarily of object labels.

5.1 Object Labels

The DIS inventory primarily consists of a DIS label for each object in the inventory. Currently the PDS inventory includes all fully archived data sets, most pending data sets, and non-data resources. Examples of resources include Web sites, user search aids such as catalogs, jukeboxes, utility software, and even people. Addition ancillary labels exist for servers and sites. In fact, new object types can be added as needed by simply designing a new label.

A DIS label describes an object using meta-data in keyword-value format as shown in figure 1. For a data set object, the keywords and values used are primarily those already specified in the PDS standards for describing a data set in the archive. In particular, a data set label includes identification information (data_set_id and data_set_name), status information (archive_status and archive_status_note), on-line links to the data (link and volume_link), relational information (instrument_name, target_name), and text descriptions (description, terse_description.) The node_name keyword identifies discipline nodes that distribute the data set and curating_node_id identifies the node responsible for creating the DIS label. It is important to note that for an archived data set, the description keyword in the DIS label simply has a link to an entry in the data set catalog. When entering the data set catalog, the data set description is the first information displayed.

```
OBJECT = DATA_SET;
 LABEL_HISTORY_NOTE = 1998-04-3 DSUPD LN:AY:DN;
 DATA_SET_NAME = VO1/VO2 MARS VISUAL IMAGING ...
 DATA_SET_ID = VO1/VO2-M-VIS-2-EDR-BR-V2.0;
 ARCHIVE_STATUS = ARCHIVED;
 ARCHIVE_STATUS_NOTE = NONE;
 DATA_SET_DESC = http://pds.jpl ...
 DATA_SET_TERSE_DESC = http://pds.jpl ...
 DATA_OBJECT_TYPE = IMAGE;
 START_TIME = 1976;
 STOP_TIME = 1980;
 NODE_NAME = IMAGING;
 CURATING_NODE_ID = IMAGING;
 TARGET_NAME = DEIMOS, MARS, PHOBOS, STAR;
 MISSION_NAME = PRE-MAGELLAN, VIKING;
 INSTRUMENT_HOST_NAME = VIKING ORBITER 1 ...
 INSTRUMENT_NAME = VISUAL IMAGING SUBSYSTEM ...
 VOLUME_ID = VO_1001, VO_1002,...
 KEYWORDS = IMAGING, MARS, VIKING, VO_1001, VO_1002 ...
 LINK = OFFLINE;
 VOLUME_LINK = ftp://pdsimage.wr... /cdroms/vo_1001/,
               ftp://pdsimage.wr... /cdroms/vo_1002/, ...
END_OBJECT;
```

Fig. 1 - Data Set DIS Label

The resource label identifies non-data resources that are available from one or more nodes. Again as seen in figure 2, identification information, on-line links, status, and a text description are provided. The status can be used to notify users of any planned changes in status or current conditions that might interest the user.

```
OBJECT = RESOURCE;
 LABEL_HISTORY_NOTE = 1998-01-01 GEO, ...
 NAME = Mars Navigator;
 DESC = The Mars Geoscience Navigator provides the capability
     to locate, display, download, and order geoscience data
     products from Mars missions.;
 NODE_NAME = GEOSCIENCES;
 CURATING_NODE_ID = GEOSCIENCE;
 STATUS = UP;
 DATA_SET_NAME = VO1/VO2_MARS_VISUAL_IMAGING_SS ...
 KEYWORDS = CAMERA, DEIMOS, IMAGE, IMAGING, MARS, ...
 LINK = http://wundow.wustl.edu/marsnav/;
END_OBJECT;
```

Fig. 2 - Resource DIS Label

A site label is created for each on-line site within the system. As seen in figure 3, this label provides a site identifier, location information, a link, and a description. Also, since there is a status and since the links provided in data set or resource label contains site identifiers, the status of the site can be merged with a link and displayed as a result of a user query.

```
OBJECT=SITE;
NAME=pdssbn.astro...;
DESC=PDS Small Bodies Node Web Server;
SERVER_HOST_ID=129.2...;
SERVER_HOST_NAME=pdssbn.astro...;
STATUS=UP;
LINK=http://pdssbn.astro.../;
END_OBJECT;
```

Fig. 3 - Site DIS Label

5.2 Object Label Creation

The labels for archived data sets as shown in figure 1 are created automatically from information contained in the data set catalog database. In particular, the relational information such as instrument_name and target_name are produced by simple joins. The keyvalues keyword however, is a collection of values from any keywords in the label that could be useful for searching.

The labels for pending data sets and resources are created manually by the curating node. However, for resource labels, the values for the keyvalues keyword are augmented automatically by the system. Using the identifiers provided for the data_set_id keyword, a utility extracts all related information from the data set catalog and augments the values for the keyvalues keyword.

5.3 Object Class Definition

The object classes are defined using a special DIS label. Figure 4 illustrates the class definition for the resource object. Each keyword has a text definition, a data type, sequence number for results ordering, and additional formatting information.

```
CLASS = RESOURCE;
 OBJECT = The object element identifies the objects class (type).
   CHAR  10  F       STD;
 NAME = The name element provides an identifier for the RESOURCE.
   CHAR  20  FSM   STD;
 DESC = The desc element provides the description of the RESOURCE.
   TEXT  30  FSM   FREE;
 LINK = The link element provides the HTTP protocol string for ...
   TEXT  41  FSM   LINK:PROMPT  Resource Online;
 KEYVALUES = The keyvalues element identifies target, missions, ...
   CHAR  70  KFS   STD;
 LABEL_REVISION_NOTE = The label_revision_note element provides ...
   TEXT  92  F       STD;
 LABEL_LINK = The label_link element provides the HTTP protocol ...
   TEXT  93  FSM  LINK:PROMPT  Source Label;
...
END_CLASS;
```

Fig. 4 - Resource Object Class Definition Label

5.4 DIS System Architecture

The current implementation of the DIS is based on the Meta-Data List Server (MEDALIS) engine, a Common Gateway Interface (CGI) script written in PERL. For the PDS implementation the engine has been named pdsserv. As seen in figure 5 the pdsserv module accesses three databases, object.db which contains the DIS labels, class.db which contains the class description labels, and system.db which contains system information. As a cgi-script, the module accepts a query in the form of an HTTP protocol compliant message. This allows users to query the module directly through the location field in their favorite browser.

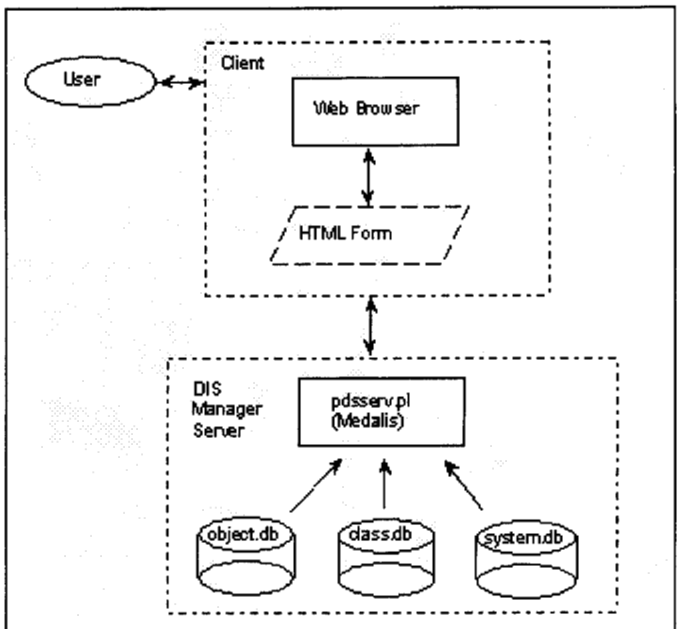

Fig. 5 - DIS Manager System Architecture

The components of the central DIS manager are a pdsserv module, class.db, system.db, and an object.db that contains a complete inventory of the PDS, as is illustrated in figure 5. At each of the discipline nodes, the components are identical except that the object.db database contains only the labels for objects local to that node. To update the DIS manager database, a polling script is periodically run to query each node for its inventory and the results are concatenated and processed to produce the object.db for the DIS manager.

5.5 The Server Engine

The pdsserv module, (MEDALIS engine), accepts queries in the form of HTTP protocol compliant queries. The two primary requirements for this query engine are the ability to constrain the query using any keyword in any label in the object.db database and that the

value of any keyword can be returned. Figure 6 provides a synopsis of a MEDALIS query. As can be seen, constraints are specified using any number of keyword=value clauses. The values, which can include regular expression components, are compared with the values of the specified keywords for every label in the object.db database to produce the result set.

SYNOPSIS

http://pds/cgi-bin/pdsserv.pl?
KEYWORD=VALUE* * => zero or more
&RETURN=KEYWORD*
&FORMAT=FORMAT** ** => zero or one.

CLAUSES

KEYWORD=VALUE - search constraint
RETURN=KEYWORD - KEYWORD selection (default->all)
RETURN=HELP - returns OBJECT class description
RETURN=DATA - redirects to LINK if query result is one
FORMAT=FORMAT - format specification
& - AND logical connective
|, !, ( ) - OR, NOT, and nesting available when encoded

FORMATS

FORMAT=HTML -> HTML list format - Default
FORMAT=HTML:M -> HTML list format, minimum keywords
FORMAT=TABLE -> HTML table format
FORMAT=PLAIN -> ODL text format
FORMAT=DISDB -> DIS database dump format

Fig. 6 - Query Synopsis

Similarly, the keyword values to be returned are specified using any number of return=keyword clauses. If no return clauses are given, all keywords in the selected labels are returned. An optional format clause allows the results to be returned in an HTML list format, HTML table format, semicolon delimited table format, or DIS label format. The logical connectives, AND, OR, NOT and parenthetic nesting are allowed. Entering HELP as a query returns the query synopsis.

Being object-based, the MEDALIS engine allows the return of the three modes of an object, the class description, object description, and the object itself. Since the DIS label is itself the object description, a standard query returns full or partial descriptions of one or more objects. The class description of an object is returned by including the clause RETURN=HELP. Note that the returned objects would have to be of the same type for this query to be valid. Finally, the actual object being described by the label can be returned by appending RETURN=DATA to the query. This clause causes a redirection to the location specified in the LINK=location statement in the label. This displays the information at that link or what the label designer considers to be the actual object.

Figure 7 shows several MEDALIS queries. The first query specifies that only data set labels that have "VG" as the first characters of the data_set_id are to be returned. For the PDS inventory, all labels describing VOYAGER data sets would be returned. The second

query would return all JUPITER related data sets by searching the keyvalues keyword. Note that the first query represents a database attribute search capability where the values of a single keyword are being searched. The second query searches the values of the keyvalues keyword – a concatenation of values from several keywords - and is more flexible. However the search is still constrained to a controlled set of values.

1) pdsserv.pl?OBJECT=DATA_SET&DATA_SET_ID=VG
returns Voyager data sets

2) pdsserv.pl?OBJECT=DATA_SET&KEYVALUES=JUPITER
returns Jupiter data sets

3) pdsserv.pl?OBJECT=RESOURCE&FORMAT=TABLE
returns resources in table format

4) pdsserv.pl?OBJECT=DATA_SET&RETURN=HELP
returns data sets class description

5) pdsserv.pl?OBJECT=DATA_SET&DATA_SET_ID=DIM&RETURN=DATA
redirects to LINK

Fig. 7 - Example Queries

The third query simply returns a list of all RESOURCES in the inventory in HTML table format. The fourth query, by specifying RETURN=HELP returns the class description for data set objects. The final query searches for a data set label that contains "DIM" in its data_set_id but redirects the server to return the information at the location specified in the LINK keyword of the DIM data set label.

In figure 8, one of the six results from the first query in figure 7 is displayed in HTML list format. Not all keywords in the data set label are shown in the result since the class description for the data set object has specified a limited set of keywords to be returned as the default. DATA_SET_NAME provides identification information and the ARCHIVE_STATUS specifies that the data set fully archived. Three link keywords in particular should be noted. The LINK keyword provides the location for what the label creator considers the entire data set. The data set in this case does not exist as a single entity that can pointed to - such as a Web page - so is considered offline. The VOLUME_LINK keyword however provides the location of 24 on-line volumes. The site status has been determined by accessing the appropriate SITE label and merging in the value of the STATUS keyword. Finally, the LABEL_LINK is a link that can be used to force the engine to return the entire label.

```
DATA_SET

    DATA_SET_NAME = VG1/VG2 JUPITER IMAGING SCIENCE SUBSYSTEM EDITED EDR ...
    DATA_SET_TERSE_DESC = This data set contains compressed level-2 (unprocessed) images from
      the Voyagers 1 and 2 encounters with Jupiter. It also contains documentation, software, and index
      directories to support access to the image files in the volume set.
    LINK = OFFLINE
    ARCHIVE_STATUS = ARCHIVED
    NODE_NAME = IMAGING
    VOLUME_LINK = Volume Online : vg_0006 - site:www-pdsimage.jpl.nasa.gov - site_status:UP
                  Volume Online : vg_0007 - site:www-pdsimage.jpl.nasa.gov - site_status:UP ...
    LABEL_LINK = Source Label - site_status:UP
```

Fig. 8 - Query Results

5.6 DIS Interfaces

Developers can create HTML forms as client interfaces to the pdsserv module. A very simple interface is illustrated in figure 9. This form allows the user to constrain queries for either data sets or resources, choose a format for the results, and accepts a list of search keywords. The query produced by this form searches the keyvalues keyword in each label.

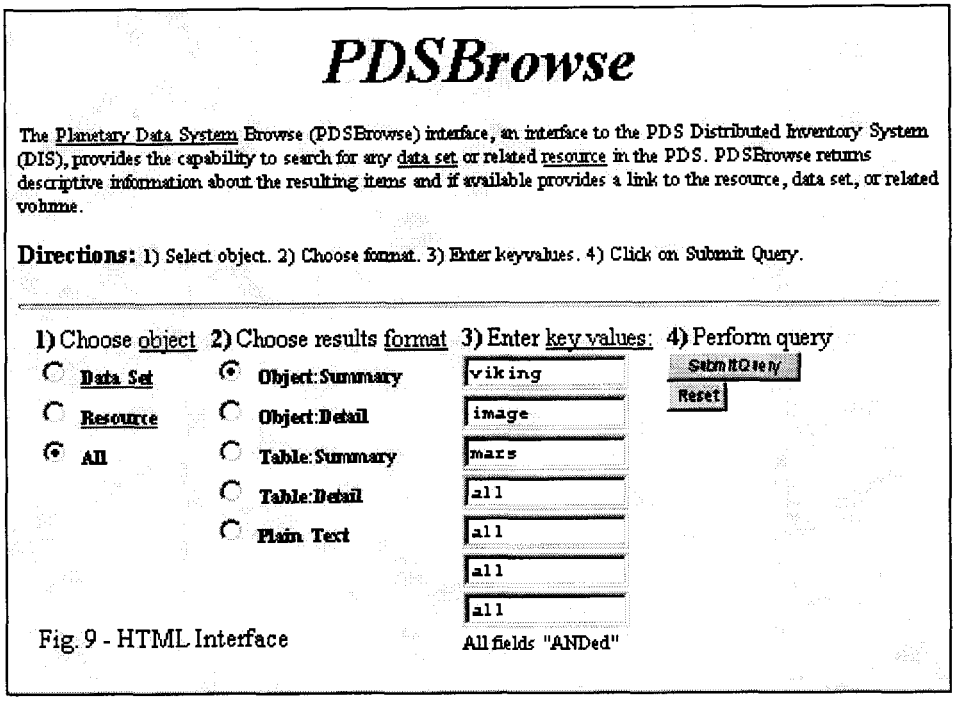

Fig. 9 - HTML Interface

6.0 PDS Access Hierarchy

As mentioned previously, the majority of the PDS archive exists on CD-ROM media. Proceeding with a library metaphor, this represents the corpus of a traditional library. However, in addition to the actual documents, the meta-data necessary for identifying,

searching, describing, and using the documents has also been packaged with the documents and placed in the library.

By placing the archive resident on CD media into a jukebox and making it accessible from the Internet, the PDS becomes a digital library accessible by the science community, the educational community, and the general public. In figure 10, this primitive level of access is represented as the base of the figure. In some cases, the CD content is moved to disk for improved efficiency.

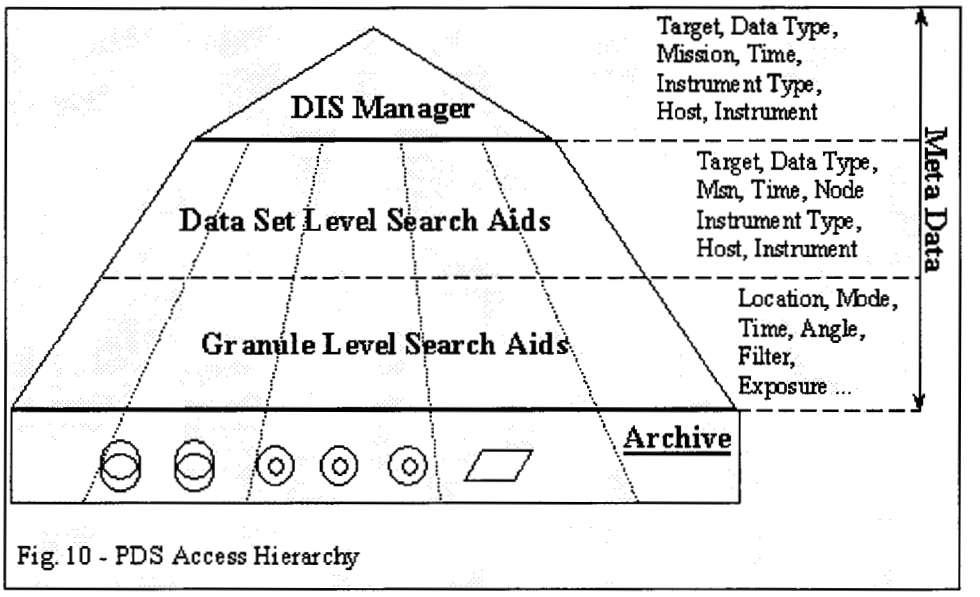

Fig. 10 - PDS Access Hierarchy

At the next level - for popular data sets and when resources are available - granule level search aids are developed. For example, the Planetary Image Atlas, a search aid developed by the PDS Imaging node, allows for users to search for images within selected image data sets. This is accomplished by extracting the meta-data stored with each image in the Viking Orbiter Mars Image data set and loading it into a relational database system. Web interfaces are then customized to allow the search for any image by using map and forms based interfaces. In fact, for the Viking data set, a user could compose a complex query for any of the approximately 30 attributes associated with an image, such as camera filter, center point location, or exposure duration.

At the data set level, the data set catalog allows for the search of archived data sets using more global attributes such as data set start_time, stop_time, data_set_id, and data_set_name. Relationships to spacecraft, instruments, missions, and targets can also be used to identify data sets.

As figure 10 shows, the DIS manager tops off the hierarchy. It is dependent on the data set catalog for the inventory of archived data sets and the discipline nodes for the inventories of pending data sets and resources.

7.0 Current Status and Future Directions

The DIS has met the three original requirements for the system. The current DIS system is operational with labels for all archived data sets, pending data sets, and available resources. Through the simple HTML form interface, the system is in constant use by scientists, the general public, discipline node personnel, and PDS management. We have several discipline node components up and are implementing the automatic update process. It has been observed that the frequency of update at some of the nodes does not warrant automatic upload, so some updates are simply e-mailed to the central node for directly updating the DIS manager database. This flexibility is allowed. Depending on available funding, more sophisticated user interfaces will be built and modifications will be made to the pdsserv engine to make it more efficient.

Other uses have also been found for the DIS. For example, in PDS operations it has been useful for reconciling the data set catalog and individual node inventories. Management is also considering producing status reports for the data set ingestion queue using archive_status values.

A simple data set granule or product server could also be implemented by creating labels from meta-data stored with the products. This would allow a user or program to access a product by simply specifying the data_set_id and product_id. Since the product type is included in the meta-data, existing utilities could be integrated to allow the transformation and display of the products.

The DIS is also being considered as a component for building interoperability between PDS and other science data systems within the NASA community. The DIS, using standard Web protocols, can be queried from external systems to locate on-line data and resources within the PDS. The DIS is also being considered as a lightweight inventory solution for other distributed data systems.

Future directions focus on staying compatible with advances in digital library technology. In particular, the developers are considering the use of XML as the language for the DIS label. The availability of XML editors will preclude the need for developing smart editors for the current DIS labels. In addition, DIS query results in XML format would be more portable, an important requirement for system interoperation. Query results in XML format can also handle more complex results and can be displayed in more sophisticated formats.

The MEDALIS engine has been converted to JAVA and a graphical interface has been developed. This is currently being tested and will soon be made operational.

8.0 Conclusion

The PDS Distributed Inventory System (DIS) provides a first level of interoperability across a heterogeneous, autonomous, and distributed data system. The DIS is a lightweight solution that was developed using standards Web protocols, a simple query

language, and labels that describe objects using simple keyword-value statements. It provides the user with the ability to search and display any object in the inventory. Query results provide users with descriptions of the objects and links on-line locations. On-line status information is also displayed.

This system, even though significant on its own as a simple yet powerful tool for developing interoperability across distributed sites, is also an important case study in the development and management of meta-data for a digital library. It is primarily because of the availability and consistency of the meta-data in the archive that this system was a success. The populating of the database simply consisted of extracting meta-data from the archive and reformatting it into the DIS label format. The success of this system is strongly tied to the standards architecture developed for and adhered to by the PDS.

9.0 Acknowledgments

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

The authors wish to acknowledge the participants of the original e-mail discussion that led to the development of the DIS. In particular Elizabeth Duxbury (PDS Image Node) and Todd King (PDS PPI Node) are prime contributors. We also wish to acknowledge the PDS staff in general and Mike Martin in particular for continued development of the PDS standards architecture.

10.0 References

[1] Russell, C. T., et al., Special Issue: Planetary Data System, Planetary and Space Science, Pergamon, Vol. 44, No. 1, 1996.

[2] Arvidson, R.A., et al., Issues and Recommendations Associated with Distributed Computation and Data Management Systems for the Space Sciences, National Academy Press, 1986.

[3] Planetary Data System Data Preparation Workbook, JPL Internal Document, JPL D-7669, Part 1, Jet Propulsion Laboratory, 1993. Also accessible at http://pds.jpl.nasa.gov/prepare.html .

[4] Planetary Data System Standards Reference, JPL Internal Document, JPL D-7669, Part 2, Jet Propulsion Laboratory, 1995. Also accessible at http://pds.jpl.nasa.gov/prepare.html .

[5] Planetary Science Data Dictionary Document, JPL Internal Document, JPL D-7116, Jet Propulsion Laboratory. Also accessible at http://pds.jpl.nasa.gov/prepare.html .

[6] Hughes, J.S., Li, Y.P., The Planetary Data System Data Model. Proceedings of Twelfth IEEE Symposium on Mass Storage Systems, 1993, 183-189.

[7] Hughes, J. S., McMahon, S. K., The Planetary Data System – A Case Study in the Development and Management of Meta-Data for a Scientific Digital Library, Lecture Notes in Computer Science, Second European Conference on Research and Advanced Technology for Digital Libraries, 1998, 335-350.